Universal fault tolerant quantum computation on bilinear nearest neighbor arrays

Ashley M. Stephens[1,*], Austin G. Fowler[2] and Lloyd C. L. Hollenberg[1]
[1]*Centre for Quantum Computer Technology, School of Physics*
University of Melbourne, Victoria 3010, Australia.
[2]*Institute for Quantum Computing*
University of Waterloo, Ontario N2L 3G1, Canada.

(Dated: May 9, 2008)

Assuming an array that consists of two parallel lines of qubits and that permits only nearest neighbor interactions, we construct physical and logical circuitry to enable universal fault tolerant quantum computation under the $[[7, 1, 3]]$ quantum code. A rigorous lower bound to the fault tolerant threshold for this array is determined in a number of physical settings. Adversarial memory errors, two-qubit gate errors, and readout errors are included in our analysis. In the setting where the physical memory failure rate is equal to one-tenth of the physical gate error rate, the physical readout error rate is equal to the physical gate error rate, and the duration of physical readout is ten times the duration of a physical gate, we obtain a lower bound to the asymptotic threshold of 1.96×10^{-6}.

PACS numbers: 03.67.Lx

I. INTRODUCTION

Quantum computation will potentially enable the efficient solution of computationally difficult problems such as factorization [1], database searching [2] and quantum system simulation [3]. However, for any quantum computer system to be feasible it must permit the scalable implementation of fault tolerant quantum error correction (FTEC) [4, 5, 6, 7]. FTEC will exact a reduction in the rate of effective gate failure with each recursive level of encoding provided that the underlying physical failure rate is below some threshold [8, 9]. The specific value of this threshold is influenced by the chosen quantum code, and the efficiency of circuits devised to implement this code, and by the spatial and operational limitations of the physical qubit array.

Given the considerable interest in quantifying the experimental requirements associated with large-scale quantum computation, a great number of numeric and analytic threshold estimates have been undertaken [10]. Typically, threshold estimates have permitted unlimited range interaction between arbitrary pairs of qubits. In reality, controlled long-range interactions of this type will be difficult to achieve. As many quantum computing proposals are based upon short-range interactions, a more valid assumption is that only nearest neighbor (NN) interactions are available [11]. The prevalence of linear nearest neighbor (LNN) proposals also suggests that a restriction may apply to the dimensionality of the qubit array in practice.

Though a threshold has been shown to exist for a one-dimensional array with next nearest neighbor interactions [12], the viability of a LNN system remains an open question. The primary difficulty associated with a strict LNN array is that a single fault in one of the SWAP gates used to interleave adjacent logical qubits prior to a transversal interaction can impart more than one error to a single logical qubit, violating the requirements of FTEC under a distance-3 quantum code. One obvious remedy for this problem would be to implement a larger distance quantum code, such as the $[[25, 1, 5]]$ Bacon-Shor code [13], the $[[23, 1, 7]]$ Golay code [14], or the [[11,1,5]] code [15], which is the most compact known distance-5 code. In addition, the use of a LNN array does not preclude the implementation of a large quantum algorithm [16]. However, due to the increase in the number of qubits and the complexity of the circuitry required to detect and correct multiple-qubit errors, it is expected that such a restriction will incur a significant threshold penalty [17]. Other ideas under investigation include a two-dimensional NN array, for which a favorable fault tolerant threshold has been obtained [18], or systems with a mechanism for qubit transport [19, 20], though there is no clear path to the near-term fabrication of any such architecture.

In this paper we consider a quasi one-dimensional array that consists of two parallel lines of qubits and permits only NN interactions. A bilinear architecture of this kind appears to be more tractable than any two-dimensional architecture and does not prohibit universal fault tolerant quantum computation under a distance-3 quantum code. Such an architecture could be realized, for example, in a donor-based system, as was originally proposed in [21], or using superconducting technology [22]. We note that a CNOT threshold has been estimated for a NN array that increases in length *and* width with each level of concatenation [23]. In this paper we present physical and logical circuitry to achieve universal fault tolerant quantum computation on a more constrained bilinear array. In addition, we determine a rigorous lower bound to the threshold for universal fault tolerant computation on

[*] electronic address: a.stephens@physics.unimelb.edu.au

this array in a number of physical settings. In the setting where the physical memory failure rate is negligible, we obtain a lower bound to the threshold that is an order of magnitude higher than that presented in [23].

The paper is organized as follows. In Section II we review the chosen FTEC protocol and the method for threshold estimation. In Section III we describe the adaptation of non-local circuitry for the fault tolerant implementation of a universal set of quantum gates to physical and logical circuitry on a bilinear NN array. Section IV contains the calculation and presentation of threshold conditions for this array. Section V concludes with a summary of results and a description of further work.

II. FAULT TOLERANT ERROR CORRECTION

The seven qubit [[7, 1, 3]] code [24] is sufficient to protect against an arbitrary single qubit error and against multiple errors with non-zero probability of success and is an attractive code for concatenation as a number of fault tolerant logical gates can be implemented transversally. For example, as for any CSS code, a logical CNOT is realized by the parallel application of a CNOT from each qubit in the control logical block to each corresponding qubit in the target logical block. The set of logical single qubit gates $\{X, Z, H, S\}$ can also be implemented in this fashion. This property leads to a reduction in the complexity of a number of fault tolerant gates relative to other CSS codes and to non-CSS codes where teleportation may be required. Furthermore, though resource requirements per logical qubit are slightly higher under the [[7, 1, 3]] code than under a block code [25], under the [[7, 1, 3]] code it is possible to operate on each logical qubit simultaneously, potentially enabling faster computation.

The most gate- and time-efficient method of syndrome extraction under the [[7, 1, 3]] code is the *encoded block* method [26]. This method requires a seven qubit ancilla block to be prepared in the [[7, 1, 3]] encoded state $|0_L\rangle$. The sequential application of transversal CNOT gates between the ancilla block and the data block allows the extraction of Z- and X-type syndrome information required for FTEC. For example, the X-type syndrome is extracted by applying a transversal CNOT with the ancilla block as the control and the data block as the target. This gate will copy any Z error on the data block to the corresponding location on the ancilla block. Measurement of the ancilla block followed by a classical parity check will determine the location of a single Z error.

If the circuit for ancilla preparation is not fault tolerant, a single fault during preparation of the state $|0_L\rangle$ may lead to more than one error being copied to the data block. For example, during X-type syndrome extraction any X errors in the ancilla block will be copied to the data block. While a probabilistic procedure of ancilla verification is typically employed to restore fault tolerance, an alternative procedure has been proposed in which ancilla verification is replaced by a decoding circuit that is applied after the transversal CNOT between the data and ancilla blocks [27]. Though a single fault during ancilla preparation may still lead to a multiple-qubit error being copied to the data block, the decoding circuit is designed such that it is possible to determine the type and location of this error. A suitable recovery operation will take the form of a product of single qubit X and Z operations. FTEC using ancilla decoding is attractive as the qubits and circuitry that were required to verify the ancilla are no longer necessary. The circuit to implement X-type syndrome extraction using the encoded block method and ancilla decoding is shown in Fig. 1.

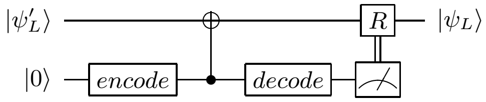

FIG. 1: X-type syndrome extraction using the encoded block method and ancilla decoding instead of verification. R is a recovery operation that may involve single qubit X and Z operations.

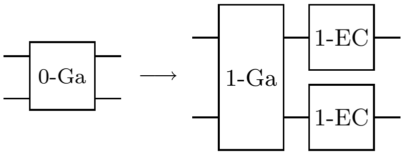

FIG. 2: To generate an encoded circuit each physical location (0-Ga) is replaced by a rectangle comprising the corresponding logical location (1-Ga) followed by error correction (1-EC) of each logical qubit.

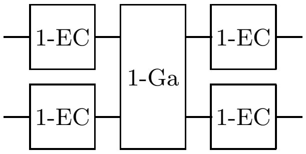

FIG. 3: Two or more faults must occur within any extended rectangle for an encoded circuit to fail.

For an architecture that can interact arbitrary pairs of qubits, to generate an encoded quantum circuit each physical qubit is replaced by a logical qubit. Each location in the original circuit (gate, memory, readout or state preparation) is also replaced by a *rectangle* [28]. Each rectangle comprises a number of physical locations and executes a fault tolerant logical operation equivalent to the location type being replaced, followed by a FTEC cycle applied to each logical qubit involved in the location. This replacement procedure is shown schematically for a two qubit gate in Fig. 2. The resultant encoded circuit will have a higher effective fidelity than the original physical circuit provided that the failure rate of the physical locations is below some threshold. To increase the fidelity of the encoded circuit, this replacement pro-

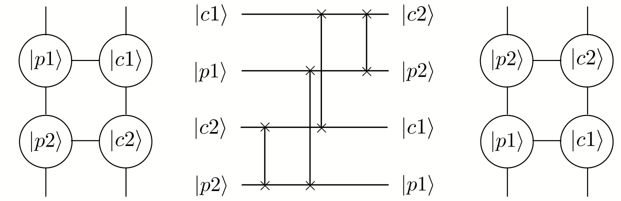

FIG. 4: The single-error SWAP of computational qubits $|c1\rangle$ and $|c2\rangle$ on a vertical bilinear array. A physical SWAP gate is only ever applied between a computational qubit and a placeholder qubit.

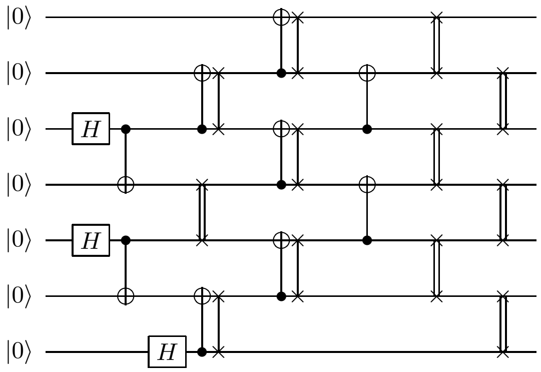

FIG. 5: Logical circuit for ancilla encoding. Compound gates are used and a double-barred SWAP symbol indicates a single-error SWAP.

cedure can be applied recursively, whereby the new physical locations are themselves replaced by rectangles. This recursive encoding forms a concatenated quantum code. Note that some technical details of this method need to be modified for the array considered in this paper.

To estimate the fault tolerant threshold an *extended rectangle* is considered [28], which is identical to a rectangle but is preceded by a FTEC cycle applied to each logical qubit. An extended rectangle is shown schematically in Fig. 3. Under the $[[7, 1, 3]]$ code, assuming that all circuits are fault tolerant, a single fault in any extended rectangle will not lead to the failure of an encoded circuit as the resultant error or errors will be correctable. Two or more faults must occur within any extended rectangle for a circuit to fail. Assuming that every combination of more than one fault will lead to failure, a lower bound to the threshold is obtained by counting the number of each type of location within the circuit of each extended rectangle, as described in detail in Section IV.

III. PHYSICAL AND LOGICAL CIRCUITRY

Adaptation of the chosen FTEC protocol to a bilinear NN qubit array involves the formulation of appropriate quantum circuits for ancilla encoding and for syndrome extraction. To achieve universal computation, circuits are also required for the fault tolerant implementation of a universal set of logical gates. One such set comprises $\{I, H, X, Z, S, S^{\dagger}, T, \mathrm{CNOT}\}$. As the [[7,1,3]] code permits the transversal implementation of each of these gates except for T, and assuming that single and two qubit gates can be compounded, the circuits required for the error corrected universal set can be adequately represented by $\{I, T, \mathrm{SWAP}\}$, where I will be referred to as a *memory* location. This is possible because the circuit for the SWAP extended rectangle comprises a similar (and slightly larger) gate layout as that for the CNOT extended rectangle. Therefore, to estimate the threshold for universal quantum computation, the construction of circuitry describing the extended rectangles of memory, SWAP, T and readout is sufficient. Note that the readout extended rectangle includes transversal readout, qubit resetting and logical state preparation.

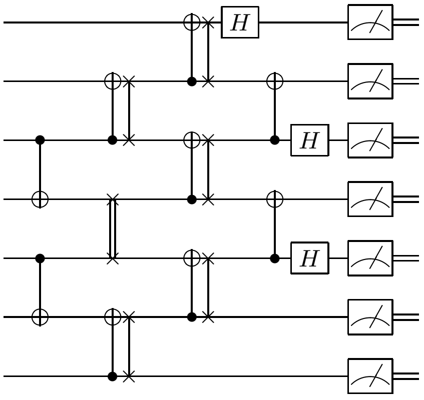

FIG. 6: Logical circuit for ancilla decoding. All measurements are made in the computational basis.

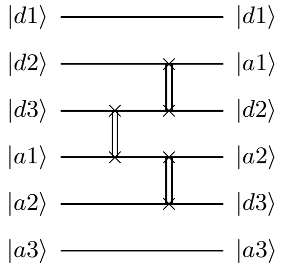

FIG. 7: Logical mesh circuit. The corresponding unmesh circuit is the circuit reversed.

In the absence of any other fault tolerant transport mechanism, for a NN architecture to be feasible it must include a fault tolerant SWAP mechanism at the physical level and at each logical level. Under the $[[7, 1, 3]]$ code, one fault during this process should impart no more than one error to any logical qubit. To satisfy this requirement, the qubit array is divided between *computational* qubits, which include all data and ancilla qubits, and *placeholder* qubits, the states of which are inconse-

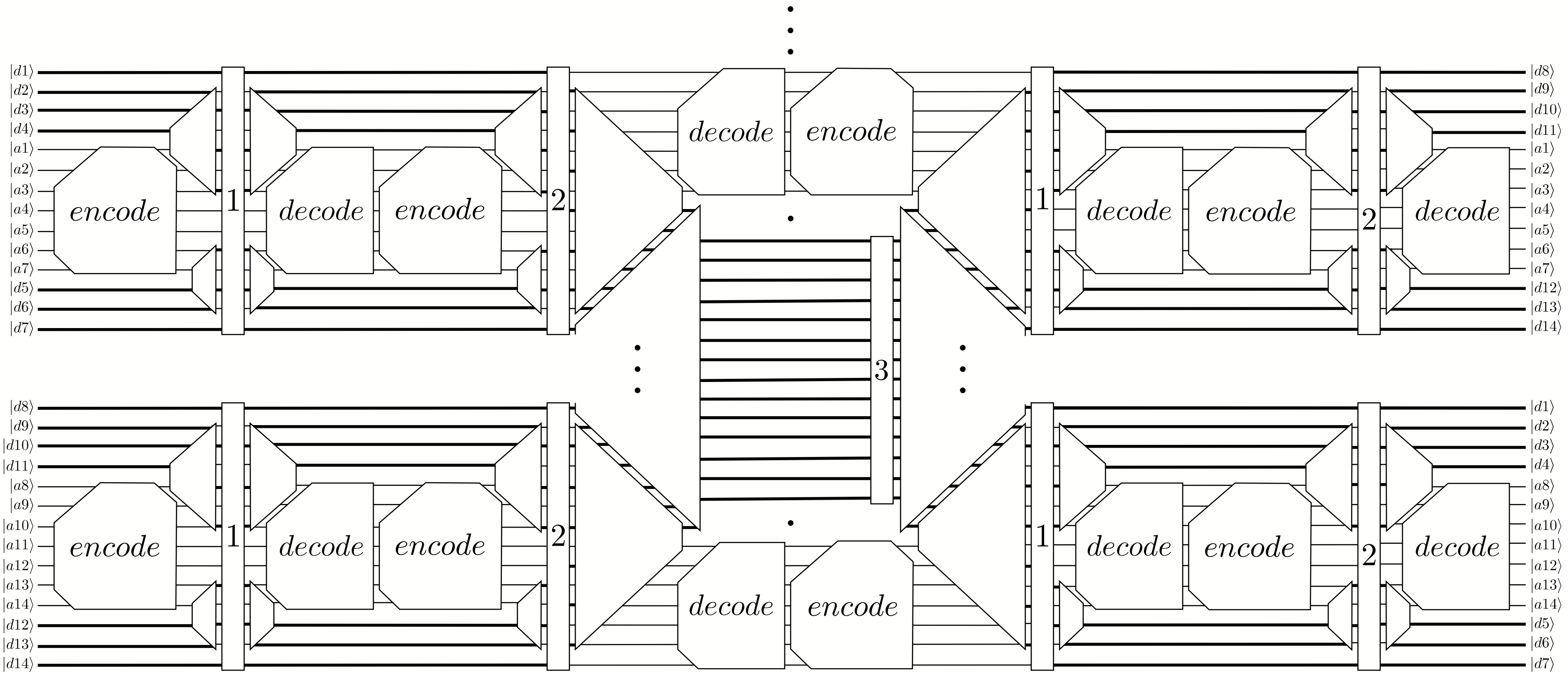

FIG. 8: Logical circuit for the SWAP extended rectangle. Bold lines represent data qubits and regular lines represent ancilla qubits. Encode and decode components refer to Fig. 5 and Fig. 6 respectively. Triangular components are appropriately sized mesh and unmesh circuits similar to Fig. 7. Rectangular components represent various transversal interactions: 1 is a CNOT with each ancilla qubit the control and each data qubit the target; 2 is a CNOT with each data qubit the control and each ancilla qubit the target, and with H applied to each ancilla qubit before and after the CNOT; and 3 is a transversal SWAP between logical data blocks. Additional SWAP gates are included in transversal interactions where required. Vertical dots indicate the presence of qubits that are not required during the CNOT extended rectangle, but are required to achieve the fault tolerant T. The central mesh circuitry is not shown explicitly but all locations are included in the data presented in Table II.

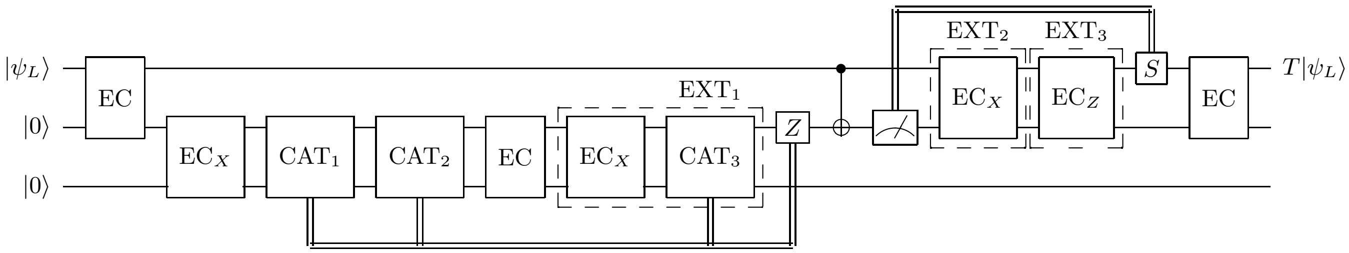

FIG. 9: Fault tolerant circuit for the T extended rectangle that is based on the implementation in Fig. 10. EC is a full error correction cycle and EC$_X$ and EC$_Z$ are partial error correction cycles that extract the X- and Z-type syndrome respectively, as shown in Fig. 1. Each CAT refers to the circuit in Fig. 11. The preparation of the state $|A_{\frac{\pi}{4}}\rangle = T|+\rangle$ is as follows: EC$_X$ will prepare the state $|0_L\rangle$ from the state $|0\rangle$, then two measurements of the operator $SX = TXT^\dagger$ are performed followed by a full error correction cycle. If an error is detected in the EC following CAT$_2$ or in either of CAT$_1$ or CAT$_2$, or if the measurement results of CAT$_1$ and CAT$_2$ disagree, then the ancilla preparation is aborted, the ancilla is reset and the additional circuitry denoted by EXT$_1$ is used. To complete the preparation, logical Z is applied to the ancilla block conditional upon the eigenvalue of the operator $SX = TXT^\dagger$ being -1. Following the interaction between the ancilla and data, logical S is applied to the data block conditional upon the associated measurement outcome being 1. If an error is detected by the second syndrome measurement within the first EC, and if EXT$_1$ has not already been applied, the additional circuitry denoted by EXT$_2$ is used. If an error is detected in EXT$_2$ the additional circuitry denoted by EXT$_3$ is used.

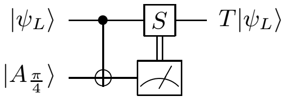

FIG. 10: Implementation of the logical rotation $T = exp(\frac{-i\pi Z}{8})$, where $|A_{\frac{\pi}{4}}\rangle = T|+\rangle$. Logical S is applied to the data block conditional upon the measurement outcome being 1. The required state $|A_{\frac{\pi}{4}}\rangle$ is prepared by the circuit in Fig. 11.

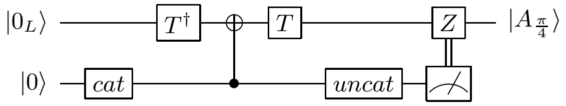

FIG. 11: Circuit to prepare the required state $|A_{\frac{\pi}{4}}\rangle = T|+\rangle$. Logical Z is applied to the data block conditional upon the measured eigenvalue of the operator $SX = TXT^\dagger$ being -1, though in the circuit in Fig. 9 this is only after the eigenvalue has been determined after two or three separate measurements. The cat state decoding procedure suggested in [27] is used to detect (but not correct) errors. Also note that a $T^\dagger T$ pair can be eliminated when this circuit is repeated in the circuit in Fig. 9.

quential to any computational processes [12]. Then, as shown in Fig. 4, the *single-error* SWAP of two computational qubits is realized through a sequence of physical SWAP gates that move two computational qubits past each other on the bilinear array. During this sequence each physical SWAP is only ever applied between a computational qubit and a placeholder qubit. As the states of placeholder qubits are inconsequential, a single faulty SWAP will impart only one error to a single computational qubit and, therefore, only one error to a single logical qubit. This forms what is effectively a SWAP-based fault tolerant transport mechanism. A single-error logical SWAP can be constructed from this underlying mechanism in order to satisfy the fault tolerant requirement at all levels. At any level above the physical level, the bilinear NN physical qubit array becomes what is effectively a LNN logical qubit array with the added property of single-error logical SWAP gates. Thus, in the formulation of circuitry describing the required extended rectangles, a distinction is made between *logical circuitry* and *physical circuitry*:

Logical circuitry. With the introduction of single-error SWAP gates, LNN circuitry can be made fault tolerant using a distance-3 code at any logical level. Logical circuitry to implement FTEC simply follows from the equivalent non-local circuitry. Figures 5 and 6 show the [[7,1,3]] ancilla encoding and decoding circuits respectively where all circuitry is valid for any level above the physical level and where a single-error logical SWAP is indicated by a double-barred SWAP symbol. Additional LNN *mesh* and *unmesh* circuits, shown in Fig. 7, are used to interleave and then separate data and ancilla blocks prior to and after X- and Z-type syndrome extraction and to interleave adjacent logical blocks prior to any inter-logical transversal interaction. Circuitry for the SWAP extended rectangle is constructed from these basic components and is shown in Fig. 8. Construction of the memory, T and readout extended rectangles similarly follows from non-local circuitry. For the T extended rectangle, our starting point was the non-local fault tolerant circuit presented in [28] which we then made more compact, reducing both the qubit and gate counts, the details of which can be found in Figs. 9-11. Finally, for the readout extended rectangle, the state $|0_L\rangle$ is prepared

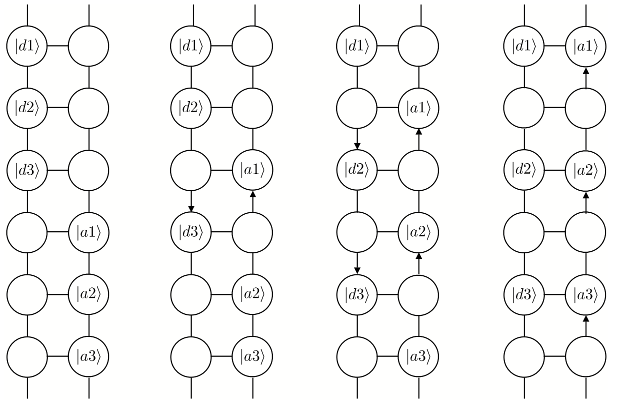

FIG. 12: An example of a physical mesh circuit that is more efficient than the circuit generated by simply replacing each SWAP in the LNN circuit in Fig. 7 with the circuit shown in Fig. 4. Four discrete steps are shown.

from the state $|0\rangle$ by the circuit in Fig. 1.

We note that a number of measurement outcomes will result in a larger T circuit, as shown in Fig. 9. For example, an additional partial or full FTEC cycle is required prior to the classically controlled S. This is to prevent the combination of an X and a Z error on different qubits in a single logical block, which may arise due to a single fault during syndrome extraction, from transforming to a combination of a Y error and an X or Z error on different qubits. To be consistent with the rigorous assumptions made in Section IV, and as our analysis requires that the depth of the T rectangle does not dependent on particular measurement outcomes, we consider the largest circuit that can result from a single fault.

Physical circuitry. To construct physical level circuitry from LNN circuitry it would be sufficient to replace every single-error logical SWAP by the single-error SWAP shown in Fig. 4. To obtain more gate- and time-efficient circuitry we note that the logical LNN mesh circuit has a much more efficient physical gate implementation, as shown in Fig. 12, than that suggested by the naive implementation shown in Fig. 4. Additional efficiency was achieved by including gates between physical qubits that are oriented perpendicular to the direction of the bilinear array. This resulted in circuitry that is significantly more efficient than that presented in [23]. All physical level circuitry was optimized by hand in this manner.

To facilitate analysis, at the physical level and at all logical levels the duration of rectangles describing all locations were arranged to permit parallel locations, as illustrated in Fig. 13. Furthermore, since, in principle, a T could occur anywhere in a circuit at any time, during encoding every physical qubit is replaced by 21 physical qubits. It is expected that the detailed arrangement of gates and qubits will ultimately, however, be an automated process. This process may involve the dense packing of FTEC cycles during memory and may also incorporate routing to avoid any defective regions that are identified during some initial characterization of the physical architecture.

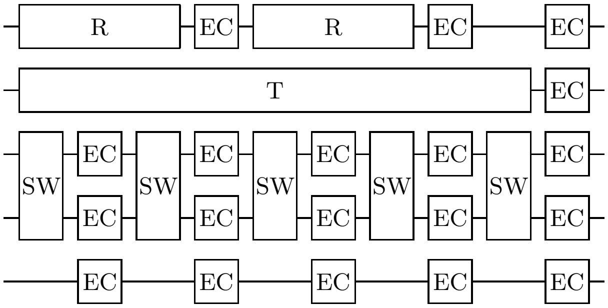

FIG. 13: Circuitry is arranged at all levels such that the duration of one T rectangle is equal to that of five memory rectangles, the duration of one readout rectangle is equal to that of two memory rectangles and the duration of one SWAP rectangle is equal to that of one memory rectangle.

IV. THRESHOLDS

To obtain our threshold estimate it was assumed that faults would occur with a probability specific to the physical location type and that this probability was equal for each location of the same type. It was also assumed that every combination of two or more faults would lead to a combination of errors that was not correctable under the $[[7,1,3]]$ code. This assumption ensures a rigorous lower bound to the threshold. Note that it is possible to find and neglect pairs of faults that lead to correctable errors, thereby relaxing this pessimistic assumption and enabling a more precise determination of the threshold. This has been demonstrated in previous work to raise the lower bound by a factor of between 3.0 and 4.5 [13, 28]. Given these assumptions, the failure probability of any physical circuit is

$$\begin{aligned} 1 &- (1-p_m)^{N_m}(1-p_S)^{N_S}(1-p_r)^{N_r} \\ &- N_m p_m (1-p_m)^{N_m-1}(1-p_S)^{N_S}(1-p_r)^{N_r} \\ &- (1-p_m)^{N_m} N_S p_S (1-p_S)^{N_S-1}(1-p_r)^{N_r} \\ &- (1-p_m)^{N_m}(1-p_S)^{N_S} N_r p_r (1-p_r)^{N_r-1}, \end{aligned} \qquad (1)$$

where p_m, p_S and p_r are the failure probabilities of physical memory, gate and readout locations respectively, and where N_m, N_S and N_r are the numbers of physical memory, gate and readout locations in the circuit. The second term corresponds to the failure of no locations. The third and fourth terms correspond to the failure of a single memory location and a single two-qubit gate location respectively, where applied single- and two-qubit errors are chosen adversarially. The final term corresponds to the failure of a single readout location, where a bit flip is applied to the classical result. Similarly, the failure probabilities of the physical level memory, SWAP, T and readout extended rectangles are given by

$$\begin{aligned} p_{1j} \;=\; 1 \;&-\; (1-p_{0m})^{N_{0mj}}(1-p_{0S})^{N_{0Sj}}(1-p_{0r})^{N_{0rj}} \\ &-\; N_{0mj}p_{0m}(1-p_{0m})^{N_{0mj}-1}(1-p_{0S})^{N_{0Sj}}(1-p_{0r})^{N_{0rj}} \\ &-\; (1-p_{0m})^{N_{0mj}}N_{0Sj}p_{0S}(1-p_{0S})^{N_{0Sj}-1}(1-p_{0r})^{N_{0rj}} \\ &-\; (1-p_{0m})^{N_{0mj}}(1-p_{0S})^{N_{0Sj}}N_{0rj}p_{0r}(1-p_{0r})^{N_{0rj}-1}, \end{aligned} \tag{2}$$

where p_{0m}, p_{0S} and p_{0r} are the probabilities of failure of physical memory, two-qubit gate and readout locations respectively and where N_{0ij} is the number of i-type locations in the physical level (level-1) j extended rectangle for $i = \{m, S, r\}$ (memory, SWAP, readout) and $j = \{m, S, T, r\}$ (memory, SWAP, T, readout). To account for the expected variation in timing of gates and readout, N_{0mj} is a expressed as a function of t_r, which is defined as the ratio of physical readout time to physical gate time. Therefore, as the relative duration of physical readout increases the number of memory locations in any extended rectangle will increase proportionately.

The failure probabilities of logical level extended rectangles can be expressed similarly, but as functions of the failure probabilities of the physical level extended rectangles, p_{1m}, p_{1S}, p_{1T} and p_{1r}, as given by Eqs. (2-5), rather than of the physical locations. For example, the failure probability of the first logical level (level-2) T extended rectangle is

$$\begin{aligned} p_{2T} \;=\; 1 \;&-\; (1-p_{1m})^{N_{1mT}}(1-p_{1S})^{N_{1ST}}(1-p_{1T})^{N_{1TT}}(1-p_{1r})^{N_{1rT}} \\ &-\; N_{1mT}p_{1m}(1-p_{1m})^{N_{1mT}-1}(1-p_{1S})^{N_{1ST}}(1-p_{1T})^{N_{1TT}}(1-p_{1r})^{N_{1rT}} \\ &-\; (1-p_{1m})^{N_{1mT}}N_{1ST}p_{1S}(1-p_{1S})^{N_{1ST}-1}(1-p_{1T})^{N_{1TT}}(1-p_{1r})^{N_{1rT}} \\ &-\; (1-p_{1m})^{N_{1mT}}(1-p_{1S})^{N_{1ST}}N_{1TT}p_{1T}(1-p_{1T})^{N_{1TT}-1}(1-p_{1r})^{N_{1rT}} \\ &-\; (1-p_{1m})^{N_{1mT}}(1-p_{1S})^{N_{1ST}}(1-p_{1T})^{N_{1TT}}N_{1rT}p_{1r}(1-p_{1r})^{N_{1rT}-1}, \end{aligned} \tag{3}$$

where N_{1iT} is the number of i-type locations in the logical level T extended rectangle for $i = \{m, S, T, r\}$. Polynomials that describe the logical level memory, SWAP and readout extended rectangles take a similar form to Eq. (6), though there are no T gates in any of these circuits. Due to the self-similarity of circuitry at all logical levels, polynomials describing level-n extended rectangles are identical to the corresponding level-2 polynomials, though they are expressed in terms of the failure probabilities of level-(n−1) extended rectangles. Furthermore, as the of duration circuitry at all logical levels is artificially set, there is no requirement for time-dependence to be included in any logical level polynomials. Tables I and II show values of N_{0ij} and N_{1ij} determined from the physical and logical circuitry presented in Section III. If memory locations are not neglected, the T extended rectangle contains the most locations at the physical level and at all logical levels. As T is required to achieve universality, the threshold for universal fault tolerant computation is given by the T threshold.

	$i=m$	$i=S$	$i=r$	depth
$j=m$	$654+28t_r$	408	40	$41+2t_r$
$j=S$	$1002+56t_r$	1122	80	$41+2t_r$
$j=T$	$3032+133t_r$	1228	128	$205+10t_r$
$j=r$	$1045+42t_r$	510	57	$82+4t_r$

TABLE I: Number of i-type locations in the physical level (level-1) j extended rectangle for $i = \{m, S, r\}$ and $j = \{m, S, T, r\}$. The depth of each rectangle is also shown where the fundamental time scale is given by the duration of a physical gate.

	$i=m$	$i=S$	$i=T$	$i=r$	depth
$j=m$	558	204	0	28	38
$j=S$	824	603	0	56	38
$j=T$	2605	619	28	98	190
$j=r$	974	255	0	42	76

TABLE II: Number of i-type locations in the logical level-n j extended rectangle for $i = \{m, S, T, r\}$ and $j = \{m, S, T, r\}$. The depth of each rectangle is also shown where the time scale is given by the duration of a level-(n−1) memory rectangle.

Following the ideas presented in [29], the level-n T threshold is where the probability of failure of the level-n T extended rectangle is equal to the probability of failure of the level-1 T extended rectangle. As any polynomial at any level can be expressed as a function of the failure probabilities of physical locations by the recursive substitution of lower level polynomials, it is possible to determine a lower bound to the level-n threshold by solving

$$p_{nT}(p_{0S}, p_{0m}, p_{0r}, t_r) = p_{1T}(p_{0S}, p_{0m}, p_{0r}, t_r), \tag{4}$$

for p_{0S}, p_{0m} and p_{0r}. To estimate a physical gate threshold error rate it is necessary to specify the ratios of the physical memory failure rate and the physical readout

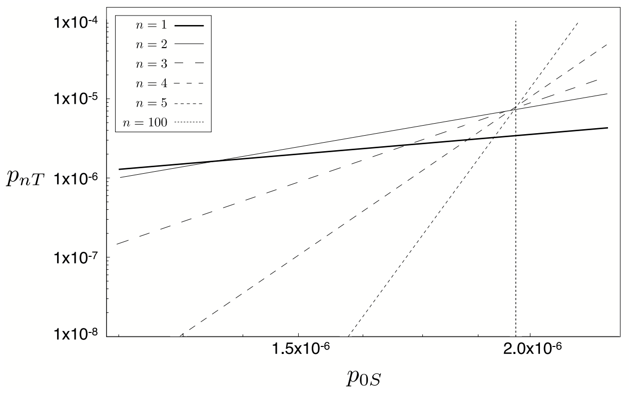

FIG. 14: $p_{nT}(p_{0S}, 0.1, 1.0, 10.0)$ for $n = \{1, 2, 3, 4, 5, 100\}$. Lower bounds to the levels 2-5 and 100 T thresholds are 1.36×10^{-6}, 1.72×10^{-6}, 1.85×10^{-6}, 1.91×10^{-6} and 1.96×10^{-6} respectively.

failure rate to the failure rate of all other physical gates, denoted by R_m and R_r respectively. The specification of R_m, R_r and t_r is referred to as a physical *setting*. Therefore, determination of a lower bound to the level-n threshold as an explicit gate failure rate in any physical setting involves solving

$$p_{nT}(p_{0S}, R_m, R_r, t_r) = p_{1T}(p_{0S}, R_m, R_r, t_r), \qquad (5)$$

for p_{0S}. Figure 14 shows the failure rate of the levels 1-5 and 100 T extended rectangles as a function of p_{0S} in the setting where $R_m = 0.1$, $R_r = 1.0$ and $t_r = 10.0$ and includes a lower bound to each of the levels 2-5 and 100 T thresholds in this setting. Levels 2-3 thresholds are particularly useful as they indicate the maximum gate failure rates tolerable in a practical quantum computer. Note that if the gate failure rate is equal to the level-n threshold, if between 2 and $n{-}1$ levels of encoding are used the encoded circuit will operate with a lower effective fidelity than at only one level of encoding. Only with levels of encoding beyond n, and, therefore, only with greater resources, will a higher fidelity encoded circuit be achieved. Realistically, gate failure rates must be at least an order of magnitude less than these thresholds to obtain significant benefit from FTEC.

To make a comparison with thresholds previously obtained for other qubit arrays it is necessary to consider the asymptotic T threshold. For a physical gate failure rate equal to this threshold recursive encoding maintains the failure rate of the logical T extended rectangle. A lower bound to the asymptotic T threshold is approximated by the level-100 T threshold, also included in Fig. 14. For a NN array that increases in length and width with each level of concatenation, in the setting where $R_m = 0.0$, $R_r = 1.0$ and $t_r = 1.0$ a CNOT threshold of 1.20×10^{-7} has been obtained [23]. In the identical setting, for a bilinear NN array we obtain a lower bound to the asymptotic T threshold of 2.88×10^{-6}. For a two-dimensional NN array, in the setting where $R_m = 0.1$, $R_r = 1.0$ and $t_r = 1.0$, accounting for pairs of faults that lead to correctable errors, a threshold for universal computation of 1.85×10^{-5} was obtained [18]. In the identical setting, for the bilinear NN array we obtain a lower bound to the asymptotic T threshold of 2.05×10^{-6}.

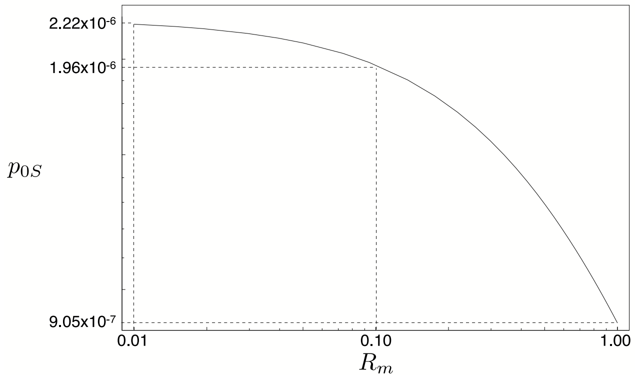

FIG. 15: Asymptotic T threshold as a function of R_m in the setting where $R_r = 1.0$ and $t_r = 10.0$.

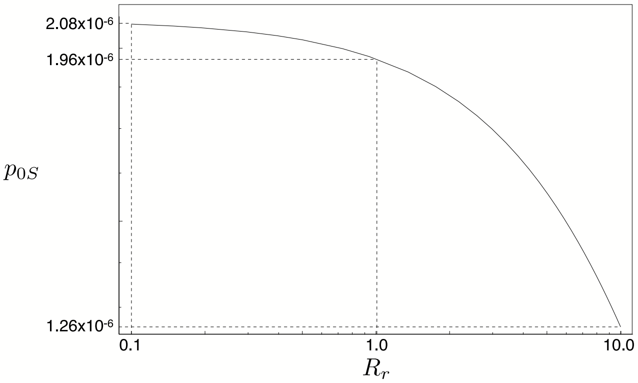

FIG. 16: Asymptotic T threshold as a function of R_r in the setting where $R_m = 0.1$ and $t_r = 10.0$.

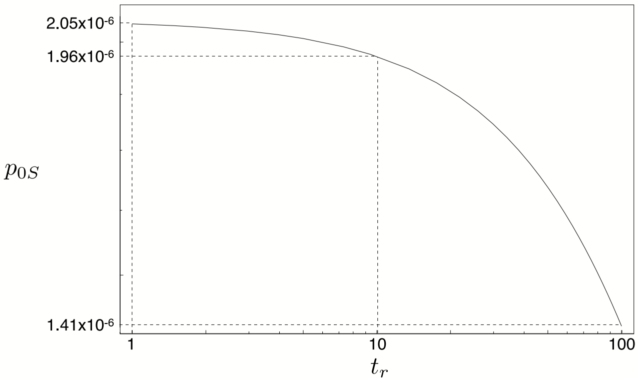

FIG. 17: Asymptotic T threshold as a function of t_r in the setting where $R_m = 0.1$ and $R_r = 1.0$.

Finally, to observe the dependency of the asymptotic threshold on the physical setting, R_m, R_r and t_r were varied from the initial setting where $R_m = 0.1$, $R_r = 1.0$ and $t_r = 10.0$, as shown in Figs. 15-17 respectively.

The greatest variation of the threshold is observed when R_m is varied. This is expected as memory locations are the predominant type in most of the physical and logical level extended rectangles. However, the overall variation from the initial threshold is not great. For example, even assuming the extreme parameters $R_m = 1.0$, $R_r = 100$ and $t_r = 1000$, the threshold is reduced by less than two orders of magnitude to 3.78×10^{-8}.

V. CONCLUSION

This paper presents physical and logical circuitry to achieve universal fault tolerant quantum computation on a bilinear NN array. The lower bound to the asymptotic T threshold derived from this circuitry is 1.96×10^{-6}. This result represents a significant improvement over a CNOT threshold previously obtained for a less constrained array [23]. We note that this improvement is despite the requirement of universality and despite the application of a more rigorous method of threshold estimation. As this improvement can be attributed to a reduction in the area of the physical and logical circuitry required to implement FTEC, it is expected that further advances in the efficiency of FTEC protocols and associated circuitry will correspond to further improvements in the threshold.

While the results presented in this paper are highly relevant to existing NN architectures and to architectures that may be restricted to NN interactions between logical qubits [21, 22], further work is required to determine the threshold for universal fault tolerant computation on a LNN array under one or more larger distance codes. Such a result would help in assessing the viability of the large number of LNN architectures under development. Further work devising specific physical schemes of long-range interaction or transport that are fault tolerant is also desirable to permit the design of architectures with higher thresholds.

AMS is grateful to Zachary Evans and Simon Devitt for helpful discussions and suggestions. AMS and LCLH are supported by the Australian Research Council, the Australian Government, and by the US National Security Agency (NSA), Advanced Research and Development Activity (ARDA), and the Army Research Office (ARO) under contract number W911NF-04-1-0290.

[1] P. Shor. *Society for Industrial and Applied Mathematics* **26**, 1484 (1997).

[2] L. Grover. *Proceedings of the* 28*th* *IEEE Symposium on the Theory of Computing*, 212 (1996).

[3] A. Aspuru-Guzik, A. Dutoi, P. Love, and M. Head-Gordon. *Science* **309**, 1704 (2005).

[4] P. Shor. *Proceedings of the* 35*th* *IEEE Symposium of Fundamentals of Computer Science*, 56 (1996)

[5] E. Knill. and R. Laflamme arXive e-print quant-ph/9608012 (1996).

[6] D. Gottesman. *Phys. Rev. A* **57**, 127 (1998).

[7] J. Preskill. In *Introduction to Quantum Computation and Information*, edited by H.K. Lo, S. Popescu, and T. Spller (World Scientific, 1998).

[8] A. Kitaev. *Proceedings of the* 3*rd* *International Conference of Quantum Communication and Measurement*, 181 (1996).

[9] D. Aharonov and M. Ben-Or. *Proceedings of the* 29*th* *Annual ACM Symposium on the Theory of Computing*, 176 (1998).

[10] see [28] and references therein. More recent threshold estimates are contained in [27] and [13].

[11] see references [11-28] in A. Fowler, C. Hill, and L. Hollenberg. *Phys. Rev. A* **69**, 042314 (2004).

[12] D. Gottesman. *J. Mod. Opt.* **47**, 333 (2000).

[13] P. Aliferis. and A. Cross. *Phys. Rev. Lett.* **98**, 220502 (2007).

[14] A. Steane. *Phys. Rev. A* **68**, 042322 (2003).

[15] D. Gottesman. arXive e-print quant-ph/9705052 (1997).

[16] A. Fowler, C. Hill, and L. Hollenberg. *Quant. Inf. Comp.* **4**, 237 (2004).

[17] A. Stephens, A. Fowler, and L. Hollenberg. (2007), in preparation.

[18] K. Svore, D. DiVincenzo, and B. Terhal. *Quant. Inf. Comp.* **7**, 297 (2007).

[19] D. Kielpinski, C. Monroe, and D. Wineland. *Nature* **417**, 709 (2002).

[20] J. Taylor, H. Engel, W. Dür, A. Yacoby, C. Marcus, P. Zoller, and D. Lukin. *Nature Phys.* **1**, 177 (2005).

[21] L. Hollenberg, A. Greentree, A. Fowler, and C. Wellard. *Phys. Rev. B* **74**, 045311 (2006).

[22] A. Fowler, W. Thompson, Z. Yan, A. Stephens, B. Plourde, and F. Wilhelm. *Phys. Rev. B* **76**, 174507 (2007).

[23] T. Szkopek, P. Boykin, H. Fan, V. Roychowdhury, E. Yablonovitch, G. Simms, M. Gyure, and B. Fong. *IEEE Trans. Nano.* **5**, No. 1 (2006).

[24] A. Steane. *Proc. R. Soc. Lond.* **452**, 2551 (1996).

[25] D. Poulin. *Phys. Rev. A* **74**, 052333 (2006).

[26] A. Steane. *Phys. Rev. Lett.* **78**, 2252 (1997).

[27] D. DiVincenzo and P. Aliferis. *Phys. Rev. Lett.* **98**, 020501 (2007).

[28] P. Aliferis, D. Gottesman, and J. Preskill. *Quant. Inf. Comp.* **6**, 97 (2006).

[29] K. Svore, A. Cross, I. Chuang, and A. Aho. *Quant. Inf. Comp.* **6**, 193 (2006).